Constructing Normalized Nonconformity Measures based on Maximizing Predictive Efficiency

Anthony Bellotti ANTHONY-GRAHAM.BELLOTTI@NOTTINGHAM.EDU.CN
School of Computer Science, University of Nottingham Ningbo China, 199 Taikang East Road, Ningbo 315100, China

Editor: Alexander Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov and Giovanni Cherubin

Abstract

The problem of regression in the inductive conformal prediction framework is addressed to provide prediction intervals that are optimized by predictive efficiency. A differentiable function is used to approximate the exact optimization problem of minimizing predictive inefficiency on a training data set using a conformal predictor based on a parametric normalized nonconformity measure. Gradient descent is then used to find a solution. Since the optimization approximates the conformal predictor, this method is called surrogate conformal predictor optimization. Experiments are reported that show that it results in conformal predictors that provide improved predictive efficiency for regression problems on several data sets, whilst remaining reliable. It is also shown that the optimal parameter values typically differ for different confidence levels. Using house price data, alternative measures of inefficiency are explored to address different application requirements.

Keywords: Conformal prediction, regression, gradient descent.

1. Introduction

Conformal predictors (CP) are predictive algorithms that are reliable, meaning they are able to output prediction sets with a guarantee of accuracy at a user-defined confidence level, assuming only that the data is distributed identically and is exchangeable (Vovk et al., 2005). Higher confidence levels are achieved at the cost of large prediction sets. Larger prediction sets are called less efficient and since confidence levels are fixed, optimization of CPs is based on maximizing predictive efficiency, in some sense. In this study, *inefficiency* of a CP is measured based on the size of the prediction set. The optimization problem can be expressed formally, but is based on step functions and so is difficult to solve directly. This study proposes an approximate method to express the optimization problem in a way that gradient descent can be used. In particular, the problem of regression in the inductive conformal prediction (ICP) framework is specifically addressed. The ICP is the CP in batch mode and mirrors the usual machine learning setting, having separate training and test sets, although additionally requiring a calibration data set (Papadopoulos et al., 2002).

Previous work to improve the efficiency of ICP typically involves developing and refining the nonconformity measure (NCM) which is central to the operation of CP and is described in Section 2 below. In particular, the development of the normalized NCM (Papadopoulos et al., 2002) was important for regression. Another appoach is to make use of quantile regression to construct CPs (Romano et al., 2019) which is also found, empirically, to provide

relatively efficient predictions. The approach taken in this study is to directly optimize a parametized normalized NCM with respect to predictive inefficiency whilst holding accuracy approximately fixed.

The objective of minimizing inefficiency measured as the mean width of prediction intervals subject to an acceptable user-defined accuracy level has been studied by Khosravi et al. (2011) and Pearce et al. (2018), but outside the context of CPs. The latter authors refer to this as the HQ (high-quality) principle for generating prediction intervals. In particular, Pearce et al. (2018) propose an approximate differentiable loss function to allow the use of gradient descent to find a minima. They show that this approach performs well on several data sets. However, their methods do not have a guaranteed inherent validity as CP does. In this study, the approach used by Pearce et al. (2018) is borrowed to develop an approximation to the ICP for regression and to use it to estimate parameters for a parameterized NCM. Since we are specifically considering regression, we use the normalized NCM (Papadopoulos et al., 2002) with a linear parametric structure in the denominator and numerator. More complex structures are possible and these can be considered in future work. Indeed, Pearce et al. (2018) apply their method within a neural network framework.

Typically ICPs are built upon underlying regression algorithms. These regression algorithms will typically optimize on a loss function based on point estimates (eg mean square error) and so do not directly optimize for the efficiency of predictions. The approach proposed here allows for direct estimation of the NCM used in CP that is specifically aligned to the objective of maximizing efficiency, albeit through an approximation. Since this optimization method emulates ICP approximately but does not have guarantee of validity, we refer to it as *surrogate* conformal predictor optimization (SCPO). We can suppose that this approach will improve performance of the CP in terms of predictive efficiency. In this study, this supposition is tested using experiments on a simulated data set and on several real world data sets. It is also shown that the optimal model parameter values typically differ for different confidence levels. The notion of estimating the CP directly based on an objective function involving efficiency is rather new. Colombo and Vovk (2020) propose a method for classification using CP based on a probabilistic efficiency measure. They use an exhaustive search which is computationally expensive for a sufficiently large parameter space, but their method could potentially be modified to use gradient descent too.

Although SCPO is not inherently valid itself, it is used to estimate parameters for the NCM which is then used in an ICP which we know is inherently valid (see Figure 1). Hence the overall method is valid.

Inefficiency does not need to be measured simply as the size of the prediction set and, indeed, Vovk et al. (2016) provide a study of alternative measures. The choice of measure should depend largely on the application and use of the CP. So, for example, in applications for which we wish to avoid especially large prediction intervals, a square term may be appropriate. Using house price data, different measures of inefficiency are explored to address different business uses.

The remainder of this article is arranged as follows. The ICP for regression and its validity are introduced and the SCPO proposed in Section 2. Experimental setting and results using simulated data and multiple real data sets are given in Section 3. It is shown that the optimization method proposed here improves predictive efficiency whilst maintaining the reliability of predictions. Finally, conclusions and future work are discussed in Section 4.

2. Methodology

- Let $\mathbf{z}_1, \cdots, \mathbf{z}_n$ be an exchangeable (or i.i.d.) sequence of examples $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ with vector of v predictor variables $\mathbf{x}_i \in \mathbb{R}^v$ and response variable $y_i \in \mathbb{R}$.
- Without loss of generality, let 1 to k index training data, $k+1$ to l index calibration data and $l+1$ to n index test data, for $1 < k < l < n$.
- A nonconformity measure (NCM) is a function $A(\mathbf{x}, y) = \mathcal{A}(\mathbf{z}_1, \cdots, \mathbf{z}_k, (\mathbf{x}, y))$.
- Let $\alpha_i = A(\mathbf{x}_i, y_i)$ denote NCM for observation i.

The *inductive conformal predictor* (ICP) gives the prediction set at confidence level $1 - \varepsilon$,

$$\Gamma^{\varepsilon}(\mathbf{x}) = \left\{ y \in \mathbb{R} : \sum_{j=k+1}^{l} \mathbb{I}(A(\mathbf{x}, y) \geq \alpha_j) + 1 > \varepsilon(l - k + 1) \right\} \tag{1}$$

where $\mathbb{I}$ is the indicator function. Given that $\mathbf{z}_{k+1}, \cdots, \mathbf{z}_n$ are exchangeable (or i.i.d.), ICP predictions are valid; ie for all $i \in \{l+1, \cdots, n\}$,

$$\mathbb{P}(y_i \in \Gamma^{\varepsilon}(\mathbf{x}_i)) \geq 1 - \varepsilon. \tag{2}$$

See Vovk et al. (2005) for details and Papadopoulos et al. (2002) for ICP for regression, in particular. Consider a special case of NCM, the normalized NCM,

$$\mathcal{A}(\mathbf{z}_1, \cdots, \mathbf{z}_k, (\mathbf{x}, y)) = \frac{|y - m(\eta; \mathbf{x})|}{\sigma(\theta; \mathbf{x})} \tag{3}$$

where m and σ are parametric forms with parameter vectors η and θ respectively, such that $(\eta, \theta) = r(\mathbf{z}_1, \cdots, \mathbf{z}_k)$ and $\sigma(\theta; \mathbf{x}) > 0$, given by Papadopoulos et al. (2002). Typically we think of r as a regression algorithm that estimates the parameters based on the training data. So in previous studies, m and σ are linear models, η are coefficients estimated using linear regression of y on $\mathbf{x}$ and θ are coefficients estimated using linear regression of the absolute value of residual in model m on $\mathbf{x}$, although other regression algorithms have been used such as artificial neural networks (Papadopoulos and Haralambous, 2011). The normalized NCM with this set-up have been shown to be effective for several regression problems (Papadopoulos et al., 2002; Papadopoulos and Haralambous, 2011; Johansson et al., 2014). However, the use of two separate models or predictive algorithms for the functions m and σ is a heuristic. In this article, we attempt to estimate the parameters of these two functions together by minimizing the width of the prediction interval.

It is known that for the normalized NCM,

$$\Gamma^{\varepsilon}(\mathbf{x}) = [m(\eta; \mathbf{x}) - q\sigma(\theta; \mathbf{x}), m(\eta; \mathbf{x}) + q\sigma(\theta; \mathbf{x})] \tag{4}$$

where q is the $(1-\varepsilon)$th quantile of the sample of NCMs in the calibration set, $\alpha_{k+1}, \cdots, \alpha_l$. With this prediction interval we can rewrite (2) as

$$\mathbb{P}\left(\mathbb{I}(y_i - (f(\eta; \mathbf{x}_i) - q\sigma(\theta; \mathbf{x}_i)) \geq 0)\mathbb{I}(f(\eta; \mathbf{x}_i) + q\sigma(\theta; \mathbf{x}_i) - y_i \geq 0) = 1\right) \geq 1 - \varepsilon.$$

The indicator functions $\mathbb{I}(x \geq 0)$ can be replaced with the Heaviside step function $H(x)$ since their values will only differ at a single point $x = 0$, so the LHS is equal to

$$\mathbb{P}\left(H(y_i - f(\eta; \mathbf{x}_i) + q\sigma(\theta; \mathbf{x}_i))H(f(\eta; \mathbf{x}_i) + q\sigma(\theta; \mathbf{x}_i) - y_i) = 1\right)$$

and writing as an expected value gives

$$\mathbb{E}\left(H(y_i - f(\eta; \mathbf{x}_i) + q\sigma(\theta; \mathbf{x}_i))H(f(\eta; \mathbf{x}_i) + q\sigma(\theta; \mathbf{x}_i) - y_i)\right) \geq 1 - \varepsilon \tag{5}$$

which expresses the validity of the conformal predictor with a normalized NCM.

Since accuracy is fixed by the confidence level, the optimization goal is to maximize the efficiency of the prediction set. There are several alternative approaches to measuring the (in)efficiency of a conformal prediction, such as the sum of p-values, the S criterion, which is not dependent on ϵ, or the size of the prediction set, the N criterion, which is dependent on ϵ. See Vovk et al. (2016) for a comprehensive study. In this study we focus on the N criterion. The size of the prediction set for the normalized NCM is given as the length of the prediction interval (4) which is $2q\sigma(\theta; \mathbf{x})$, so across a sequence of examples S we can set the goal to minimize the mean loss,

$$\frac{1}{|S|}\sum_{i \in S}(2q\sigma(\theta; \mathbf{x}))^p$$

for some power $p > 0$, so $p = 1$ expresses a linear loss and $p = 2$ is a square loss, in particular. This is minimized whilst ensuring that accuracy is at the required confidence level; ie ensuring (5). The inequality in (2) and hence (5) deals with the case when two or more examples have the same value of α_i. It is possible to construct NCMs where this is likely to happen, but with the normalized NCM (3), this is an unlikely outcome so long as predictor variable values are at sufficient granularity. Hence, if the probability is greater than the confidence level, it will not be by very much and (5) will be very nearly an equality. Given this, we can consider the manifestation of (5) on the sample S as the accuracy given by

$$\text{Acc}_S = \frac{1}{|S|}\sum_{i \in S} H(y_i - m(\eta; \mathbf{x}) + q\sigma(\theta; \mathbf{x}))H(m(\eta; \mathbf{x}) + q\sigma(\theta; \mathbf{x}) - y_i) \approx 1 - \varepsilon.$$

This approximation is more accurate with larger sample size in S and could be stated more precisely since the events $y_i \in \Gamma^\varepsilon(\mathbf{x}_i)$ are i.i.d and so Acc_S follows a binomial distribution around $1 - \varepsilon$. However, for the purpose of optimizing efficiency it is only necessary that Acc_S does not deviate substantially from the confidence level and so a simple square loss term can be used for this purpose. Hence combining this with the mean loss on interval size gives a loss function,

$$L_S(\eta, \theta) = \frac{1}{|S|}\sum_{i \in S}(2q\sigma(\theta; \mathbf{x}))^p + \lambda(\text{Acc}_S - (1 - \varepsilon))^2 \tag{6}$$

where $\lambda > 0$ is a constant expressing the relative cost of controlling accuracy. Since we want to maintain accuracy closely, we would expect λ to be quite large. Minimizing this loss gives

$$(\hat{\eta}, \hat{\theta}) = r(\mathbf{z}_1, \cdots, \mathbf{z}_k) = \arg\min_{\eta, \theta} L_S(\eta, \theta). \tag{7}$$

This optimization problem does not express the ICP and, in particular, does not have the inherent property of validity, but is empirically valid through the loss function. It is an approximation to ICP so that by using the estimates $(\hat{\eta}, \hat{\theta})$ in the NCM for ICP, this will lead to predictions that are close to optimally efficient for the model structures given by m and d. A similar optimization problem jointly minimizing predictive efficiency along with accuracy has been studied by Pearce et al. (2018). The main differences are that (6) intends for accuracy to be a specific confidence level whereas Pearce et al. (2018) form a loss function that allows for an inequality, so accuracy can be greater than the confidence level, and (6) has the power term p. We follow a similar strategy as Pearce et al. (2018) by using a differentiable approximation of the problem so that gradient descent can be applied.

For (7) to emulate ICP, it would be ideal to set S to be an independent test set and for q to be computed on an independent calibration set since this is what ICP does. However, we should not use data $\mathbf{z}_{k+1}, \cdots, \mathbf{z}_n$ to do this, as this would mean that the estimates $(\hat{\eta}, \hat{\theta})$ are computed from these data, violating the conditions to ensure the ICP is valid. Therefore, (7) must only involve the training data. One way to do this is divide the training set into three parts just for the purposes of optimization: a proper training set, a second calibration set and a second test set. However, it is unlikely we will have sufficient data for the luxury of having so many data subsets. Hence the training data is used for all three purposes. This will mean that validity on the training set will not hold and we should expect some overfit. However, this is not a problem since the optimization will result in a fixed NCM parametized by $(\hat{\eta}, \hat{\theta})$, ie (3), that can then be used by an ICP which will be valid on the test set. The overfitting will likely mean that efficiency on the test data will be less than on the training data, but overfitting is not an unusual problem to face in machine learning.

The strategy of using training data in all roles for the purpose of optimization means (1) the sequence $S = (1, \cdots, k)$ and (2) the quantile q is from the distribution $\alpha_1, \cdots, \alpha_k$, instead of from the true calibration set. Let us call this quantile q' to express this difference. One thing to notice about the NCM is that the CP is *invariant* to changes in the scaling of the NCM. This can be seen in (1), since the prediction set is unchanged by rescaling by a constant factor, as all the NCM values will be rescaled by the same amount. For this reason, we can choose whatever scaling we want. In particular we will choose a scaling that sets $q' = 1$. Intuitively this does not seem right since q' is a function of the distribution of α_i's and this in turn is a function of (η, θ), therefore q' is also a function of (η, θ). However, changing the scaling of the NCM, by setting q' to a fixed value, has no impact on the performance of ICP: the ICP will perform the same for any value of q'. Setting $q' = 1$ means q can be dropped from Acc_S and $L_S(\eta, \theta)$. In experiments given in Section 3, we find that $q \approx q' = 1$ and the deviation of q from 1 is a consequence of the ICP adjusting for the overfitting of (7) on the training data and the size of the data sample.

The final difficulty for solving optimization problem (7) is that Acc_S is formed from a series of Heaviside step functions H, hence $L_S(\eta, \theta)$ is discontinuous at many places. To resolve this problem, we use

$$H(x) = \lim_{\gamma \to \infty} (1 + \exp(-\gamma x))^{-1}$$

and for practical purposes the Heaviside step function can be approximated by the expit sigmoid function, $H(x) \approx (1 + \exp(-\gamma x))^{-1}$ for sufficiently large γ. This leads to a new

loss function that approximates $L_S(\eta, \theta)$:

$$L(\eta, \theta) = \frac{1}{k} \sum_{i=1}^{k} (2\sigma(\mathbf{x}_i; \theta))^p + \lambda V^2$$

where

$$V = (1 - \varepsilon) - \frac{1}{k} \sum_{i=1}^{k} l_i u_i,$$

$$l_i = (1 + \exp(\gamma[y_i - m(\mathbf{x}_i; \eta) + \sigma(\mathbf{x}_i; \theta)]))^{-1},$$

$$u_i = (1 + \exp(\gamma[m(\mathbf{x}_i; \eta) + \sigma(\mathbf{x}_i; \theta) - y_i]))^{-1},$$

remembering q is replaced with $q' = 1$. This leads to the SCPO problem defined as

$$(\hat{\eta}, \hat{\theta}) = \arg\min_{\eta, \theta} L(\eta, \theta). \tag{8}$$

that approximates (7). Finally, σ is expressed in the form,

$$\sigma(\mathbf{x}_i; \theta) = \exp(d(\mathbf{x}_i; \theta))$$

to ensure that $\sigma(\mathbf{x}_i; \theta) > 0$.

Gradients can then be used in a gradient descent algorithm as follows,

$$\begin{aligned} \frac{\partial L(\eta,\theta)}{\partial \eta_j} &= -2\lambda V \tfrac{1}{k} \textstyle\sum_{i=1}^{k} \frac{\partial l_i u_i}{\partial \eta_j} \\ &= \tfrac{2\lambda\gamma}{k} V \textstyle\sum_{i=1}^{k} l_i u_i (u_i - l_i) \frac{\partial m(\mathbf{x}_i;\eta)}{\partial \eta_j}, \end{aligned}$$

$$\begin{aligned} \frac{\partial L(\eta,\theta)}{\partial \theta_j} &= \tfrac{1}{k} \textstyle\sum_{i=1}^{n} 2^p p \exp(pd(\mathbf{x}_i;\theta)) \frac{\partial d(\mathbf{x}_i;\theta)}{\partial \theta_j} - 2\lambda V \tfrac{1}{k} \textstyle\sum_{i=1}^{k} \frac{\partial l_i u_i}{\partial \theta_j} \\ &= \tfrac{1}{k} \textstyle\sum_{i=1}^{k} \left[2^p p \exp(pd(\mathbf{x}_i;\theta)) + 2\lambda\gamma V l_i u_i (l_i + u_i - 2) \exp(d(\mathbf{x}_i;\theta))\right] \frac{\partial d(\mathbf{x}_i;\theta)}{\partial \theta_j} \end{aligned}$$

These gradients can be computed so long as m and d are both differentiable. Different forms are possible, but for this study we use simple linear parameterizations, $m(\mathbf{x}_i; \eta) = \eta \cdot \mathbf{x}_i$ and $d(\mathbf{x}_i; \eta) = \theta \cdot \mathbf{x}_i$. Note that an intercept term is included using a variable that always takes the value 1 in $\mathbf{x}_i$.

To recap, validity is not a property of SCPO, but SCPO is empirically valid through minimizing V^2 on training data. We use its estimates $(\hat{\eta}, \hat{\theta})$ in the NCM to then implement the ICP using the calibration set to get predictions on the test set, as illustrated in Figure 1. The parameter q in ICP measures the difference between the SCPO and ICP: the closer q is to 1, then the closer the two predictors are. In particular, $q = 1$ means the SCPO and ICP are the the same region predictors. We expect $q > 1$ to account for overfitting in the process of fitting SCPO to just the training data.

Figure 1: Using SCPO with ICP

3. Experimental Results

We use a normalized NCM with a separate linear model to predict outcome and a separate linear model to predict log absolute residual, both using OLS linear regression, as a Baseline ICP to compare SCPO against. This structure of ICP with separate modelling for m and σ in (3) is typical in the literature for CP applied to regression.

A simple gradient descent algorithm is applied. However, it was found that performing a random change in (η, θ) every 10,000 iterations helped in reaching a good minima. The random change consisted of adding a term Jump $\times \eta_j s$ to each η_j where Jump is a hyper-parameter and s is randomly drawn from a standard normal distribution, and similarly for each θ_j. The number of times a random jump is made is called Cycles in the Results given below. As with other methods such as neural networks, data needs to be standardized prior to applying gradient descent. This is done by dividing all predictor variables through by their sample standard deviations computed on the training data set, for each experiment.

The learning rate was set to $10/(\gamma\lambda)$ in all experiments, by trial and error. Experiments suggest this may be too small since the observed change in loss from one iteration to another is small. However, it is better to have a smaller value than a larger one so that it does not overshoot the minimum.

All ICPs are evaluated by predictive accuracy (Acc) which corresponds to (5) and hence should be approximately equal to the confidence level if the ICP is valid and by predictive inefficiency (Ineff) which is measured simply as mean predictive interval width across the test data set.

We tested the proposed method on a simple simulated data set with just two predictor variables, x_1, x_2, both generated as absolute values of standard normal random numbers and one outcome variable $y = 0.5x_1 + x_2 + \epsilon$ where error term is generated as $0.5x_1$ times standard normal random numbers. Therfore the simulation is deliberately set up with heterogeneous errors and we would expect that predictive intervals will be wider for test examples with larger values of x_1. Results for different parameter settings are shown in Table 1. In all cases, $p = 1$. We see that all ICPs are valid (ie accuracy matches confidence level), the optimized ICP performs better than the baseline (ie Ch > 0) and the results are robust to different parameter settings. Table 2 shows differencies between parameters for the simulated data with settings $N = 10000$, $\gamma = 100$, $\lambda = 100$, Cycles=5, Jump=0.2. The values of η model the generating function for y and is stable across all ICPs. For θ, we see that θ_1 has a major effect compared to θ_2 and it is positive. This is what we expect given the way the error term is generated as positively correlated with x_1 and not x_2. Higher confidence levels require larger values of θ_1 to enable wider predictive intervals. The final column (Ineff at 95%) shows how the NCM formed from the estimated (η, θ) perform at the fixed confidence level 95%. Surprisingly, at least in this setting, all optimized ICPs perform approximately as well as each other.

Figure 2 shows a contour plot of $\log L(\eta, \theta)$ across a range of values of η_1 and θ_1 for the simulated training data. It shows that the loss function is well-behaved with a clear minima.

We tested this method on several publicly available data sets as described in Table 3. Most data sets are sourced from the UCI machine learning repository (Frank and Asuncion, 2010). The Ames data set was compiled by De Cock (2011) for educational purposes and

Experiment settings							Baseline ICP		SCPO ICP		
	N	$1-\varepsilon$	γ	λ	Cycles	Jump	Acc	Ineff	Acc	Ineff	Ch
1	500	0.9	100	100	5	0.2	0.916	2.015	0.898	1.590	0.211
2	1000	0.9	100	100	5	0.2	0.904	2.347	0.908	1.616	0.312
3	10000	0.9	100	100	5	0.2	0.903	2.123	0.902	1.529	0.280
4	50000	0.9	100	100	5	0.2	0.899	2.158	0.897	1.545	0.284
5	10000	0.95	100	100	5	0.2	0.955	2.449	0.952	1.890	0.228
6	10000	0.95	10	100	5	0.2	0.940	2.549	0.946	1.900	0.255
7	10000	0.95	1000	100	5	0.2	0.950	2.701	0.948	1.958	0.275
8	10000	0.95	100	10	5	0.2	0.949	2.723	0.948	1.978	0.274
9	10000	0.95	100	1000	5	0.2	0.951	2.708	0.950	1.947	0.281
10	10000	0.95	100	100	1	0.2	0.951	2.717	0.955	1.959	0.279
11	10000	0.95	100	100	1	0.2	0.951	2.851	0.953	1.980	0.306
12	10000	0.95	100	100	5	1	0.945	2.628	0.954	1.928	0.266
13	10000	0.99	100	100	5	0.2	0.989	3.561	0.993	2.715	0.238

Table 1: Results with simulated data. N = number of examples in each of training, calibration and test data sets. Ch is change ratio between inefficiency for SCPO method against baseline; ie Ch = 1 - Ineff(SCPO ICP)/Ineff(Baseline ICP).

$1-\varepsilon$	η_0	η_1	η_2	θ_0	θ_1	θ_2	Ineff at 95%
Baseline	0.0000	0.356	0.718	−1.745	0.879	−0.0124	2.557
0.8	0.0097	0.362	0.729	−0.682	0.338	−0.0186	1.958
0.9	−0.0197	0.409	0.727	−0.399	0.394	−0.0282	1.910
0.95	−0.0028	0.366	0.746	−0.278	0.415	0.0310	1.918
0.99	−0.0276	0.362	0.723	−0.191	0.473	0.0180	1.906

Table 2: Parameter estimates for ICP at different confidence levels. Baseline refers to the baseline linear regression, not SCPO. Ineff at 95% is inefficiency measure when using these parameters for prediction at 95% confidence level.

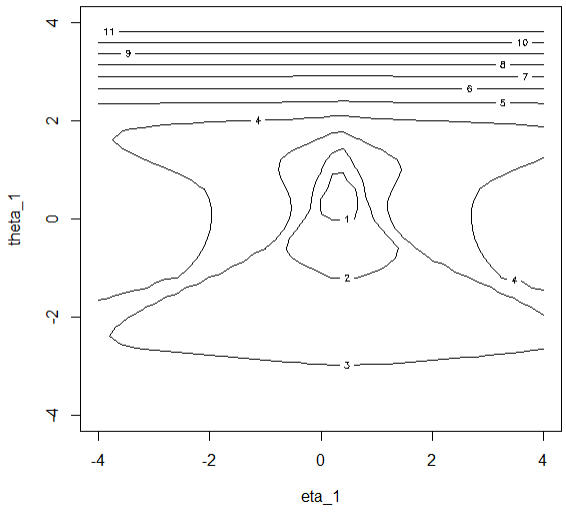

Figure 2: Contour plot of $\log L(\eta, \theta)$ across a range of values of η_1 and θ_1 for the simulated training data, for $1 - \varepsilon = 0.95$, $\gamma = 100$ and $\lambda = 100$.

the KC data set is available from the Kaggle competition site [1]. For the Ames data set, only the 9 most predictive variables are included based on stepwise variable selection using OLS linear regression. To check the robustness of the results, the experiments were repeated with different random selections of training, calibration and test sets. They give similar results from which we can draw the same conclusions.

Results at various confidence levels are shown in Tables 4 and 5. Settings were fixed at $p = 1$, $\gamma = 100$, $\lambda = 1000$, Cycles=10, Jump=0.2. For all six data sets, accuracy approximately matches the confidence level, as we would expect of ICP. In four out of the six experiments, using SCPO improves performance in terms of efficiency. For those two data sets where performance is not improved, the magnitude of difference in Ineff is relatively small (Ch ≥ -0.011). For contrast, the mean Ch over *all* six data sets is 0.063, 0.060, 0.061 and 0.149 for confidence levels 80%, 90%, 95% and 99% respectively, demonstrating an overall improvement in performance across these data sets. The tables also show that in general ICP optimized for one confidence level are not particularly good when applied for a different confidence level, contrary to results on the simulated data.

If we look at the two data sets, for which the method does not perform well, Bias and CCPP, they exhibit poor model fit for the model of absolute value of residual, σ in the normalized NCM (3) for the baseline ICP with $R^2 = 0.051$ and 0.0079 for Bias and CCPP respectively. This makes sense since it means that the denominator of the normalized NCM is not greatly influenced by the predictor variables and therefore suggests that the joint estimation of m and σ provided by SCPO is less likely to provide an advantage. However,

1. `https://www.kaggle.com/harlfoxem/housesalesprediction`

Name	*Description*	*Response y*	n	v	#Train	#Cal	#Test
Ames	US housing data	Sale price	2928	9	1464	732	732
Bias	Bias correction on temperature estimate (Cho et al., 2020)	Minimum temperature	7752	24	3000	2000	2590
CCPP	Combined Cycle Power Plant (Kaya et al., 2012)	Energy output	9568	4	4000	3000	2568
KC	US housing data	Sale price	21613	24	10000	5000	6613
GPU	GPU performance data (Nugteren and Codreanu, 2012)	Performance time (average)	241600	14	20000	20000	20000
Super	Superconductor (Hamidieh, 2018)	Critical temperature	21263	81	10000	5000	6263

Table 3: Data sets. n = number of examples; v = number of predictor variables; # refers to numbers of examples used in the training, calibration and test set respectively for ICP. For the Bias data set, 162 rows were removed due to missing values.

this is not the whole explanation since the GPU data set also has poor fit for σ ($R^2 = 0.015$) but still shows improved performance using SCPO.

Tables 4 and 5 also report the value of q which shows how close the SCPO predictor is to ICP with $q = 1$ meaning they are the same. In particular, q is a measure of the overfit of SCPO on training and its deviation from validity (ie when $V = 0$). We observe that deviation of q from 1 is small, but typically higher with higher confidence level (0.99). Also, data sets with larger sample size such as GPU, give lower values of q for all confidence level, indicating lower overfitting with larger training sample size.

Figure 3 shows how the distribution of the width of prediction intervals changes with different values of $p = 1$ for linear error and $p = 2$ for square error on interval size, as given in (6) for experiments with the Ames House Price data. Since the square error penalizes large intervals more, this reduces the size and number of outliers and tends to push the distribution of interval widths to the left, although the mean absolute error (Ineff) will be higher. We observe this in the figure: the right graph shows fewer outliers (values greater than 4) and less predictions within the range 2 to 4. However, the sample mean interval width is slightly higher at 1.377 for $p = 2$ against 1.360 for $p = 1$. This result has an immediate business interpretation: if we want to build an automated valuation model that provides a good average performance and can tolerate a large number of extremely large (and hence useless) price predictions then use $p = 1$, whereas if we like to ensure that the large majority of price predictions are within a tolerable range, whilst sacrificing average performance, then $p = 2$ can be used. The former may be useful for mortgage portfolio

Data set	Method	$1-\varepsilon$	Acc	Ineff	Ch	q
Ames	Baseline	0.8	0.792	0.841		
	SCPO	0.8	0.796	0.752	0.106	1.06
	Baseline	0.9	0.902	1.11		
	SCPO	0.9	0.889	1.047	0.057	1.05
	Baseline	0.95	0.945	1.362		
	SCPO	0.95	0.941	1.35	0.009	1.09
	SCPO (0.8)	0.95	0.951	2		
	SCPO (0.9)	0.95	0.963	1.53		
	N/a *	0.99				
Bias	Baseline	0.8	0.801	0.998		
	SCPO	0.8	0.804	1.003	-0.005	1.08
	Baseline	0.9	0.897	1.284		
	SCPO	0.9	0.897	1.286	-0.002	1.07
	Baseline	0.95	0.946	1.546		
	SCPO	0.95	0.953	1.563	-0.011	1.11
	SCPO (0.8)	0.95	0.946	1.569		
	SCPO (0.9)	0.95	0.941	1.532		
	SCPO (0.99)	0.95	0.946	1.556		
	Baseline	0.99	0.987	2.231		
	SCPO	0.99	0.988	2.158	0.033	1.24
CCPP	Baseline	0.8	0.81	0.669		
	SCPO	0.8	0.806	0.669	0.000	1.04
	Baseline	0.9	0.905	0.83		
	SCPO	0.9	0.908	0.834	-0.005	1.03
	Baseline	0.95	0.955	0.967		
	SCPO	0.95	0.96	0.97	-0.003	1.02
	SCPO (0.8)	0.95	0.954	1.058		
	SCPO (0.9)	0.95	0.959	0.988		
	N/a *	0.99				

Table 4: Results for different data sets. SCPO refers to ICP based on SCPO. SCPO(c) refers to optimizing the parameters for confidence level c. N/a * means insufficient data for confidence level 0.99. Ch is the change ratio between inefficiency for SCPO against baseline ICP; ie Ch = 1 - Ineff(SCPO)/Ineff(Baseline).

Data set	Method	$1-\varepsilon$	Acc	Ineff	Ch	q
KC	Baseline	0.8	0.808	1.202		
	SCPO	0.8	0.806	0.985	0.181	1.03
	Baseline	0.9	0.906	1.559		
	SCPO	0.9	0.895	1.311	0.159	1.03
	Baseline	0.95	0.957	1.921		
	SCPO	0.95	0.956	1.748	0.090	1.11
	SCPO (0.8)	0.95	0.948	2.035		
	SCPO (0.9)	0.95	0.948	1.815		
	SCPO (0.99)	0.95	0.956	1.804		
	Baseline	0.99	0.991	3.113		
	SCPO	0.99	0.991	2.797	0.102	1.39
GPU	Baseline	0.8	0.796	1.698		
	SCPO	0.8	0.798	1.625	0.043	1.00
	Baseline	0.9	0.896	2.175		
	SCPO	0.9	0.898	2.085	0.041	1.00
	Baseline	0.95	0.948	2.574		
	SCPO	0.95	0.95	2.319	0.099	1.02
	SCPO (0.8)	0.95	0.95	2.806		
	SCPO (0.9)	0.95	0.95	2.841		
	SCPO (0.99)	0.95	0.951	2.431		
	Baseline	0.99	0.989	3.398		
	SCPO	0.99	0.99	2.946	0.133	1.12
Super	Baseline	0.8	0.785	1.208		
	SCPO	0.8	0.786	1.146	0.051	1.00
	Baseline	0.9	0.895	1.602		
	SCPO	0.9	0.889	1.428	0.109	1.00
	Baseline	0.95	0.949	1.984		
	SCPO	0.95	0.95	1.62	0.183	1.02
	SCPO (0.8)	0.95	0.946	2.297		
	SCPO (0.9)	0.95	0.948	1.69		
	SCPO (0.99)	0.95	0.946	1.659		
	Baseline	0.99	0.991	2.901		
	SCPO	0.99	0.989	1.951	0.327	1.10

Table 5: Results for different data sets. SCPO refers to ICP based on SCPO. SCPO(c) refers to optimizing the parameters for confidence level c. Ch is the change ratio between inefficiency for SCPO against baseline ICP; ie Ch = 1 - Ineff(SCPO)/Ineff(Baseline).

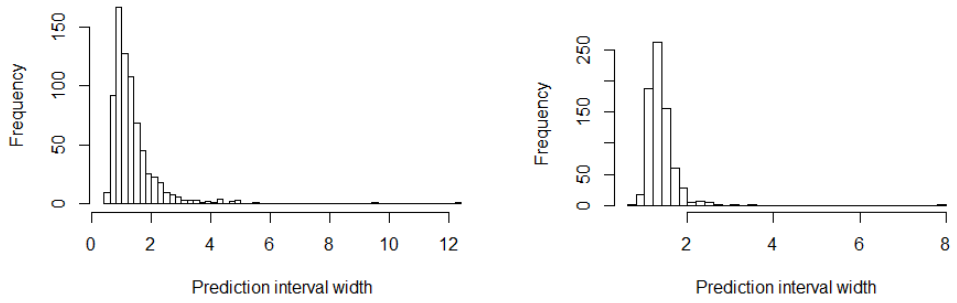

Figure 3: Histograms of predictive inefficiency (interval widths) for $p = 1$ (left) and $p = 2$ (right) in Equation (6).

risk models when the aggregate measure is important, whilst the latter may be practical for delivery to customers where each individual valuation is important. It also suggests that other loss functions for inefficiency are worth exploring.

4. Conclusion

We have shown how an optimization problem, SCPO, can be defined to minimize predictive inefficiency at approximately a fixed accuracy and hence emulate the operation of the ICP for regression to estimate an optimal NCM. When used in ICP, this NCM leads to more efficient predictions compared to the baseline ICP with the usual normalized NCM (3) with two separate models for numerator and denominator of the normalized NCM for several example data sets. For the two data sets when SCPO does not improve the predictive efficiency, it is never substantially worse with a difference of no more than 1.1% increase in mean width of prediction intervals.

The gradient descent algorithm that has been employed is rather crude (eg with a fixed small learning rate) which could prompt further research to improve the optimization of the NCM to improve computational efficiency. Also, in this initial study, a simple linear model is used to express m and d, but the method could be extended to other model structures that use gradient descent, such as artificial neural networks. This will also be a further direction for research of this method.

References

D. Cho, C. Yoo, J. Im, and D. Cha. Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas. *Earth and Space Science*, 2020.

N Colombo and V Vovk. Training conformal predictors. *arxiv*, 2020. URL https://arxiv.org/pdf/2005.07037.pdf.

Dean De Cock. Ames, Iowa: Alternative to the boston housing data as an end of semester regression project. *Journal of Statistics Education*, 2011.

A Frank and A Asuncion. UCI Machine Learning Repository. *Irvine, CA*, 2010. URL http://archive.ics.uci.edu/ml.

Kam Hamidieh. A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, 154:346–354, November 2018.

Ulf Johansson, Henrik Boström, Tuve Löfström, and Henrik Linusson. Regression conformal prediction with random forests. *Machine Learning*, 97:155–176, 2014.

H Kaya, P Tfekci, and SF Grgen. Local and Global Learning Methods for Predicting Power of a Combined Gas & Steam Turbine. In *Proceedings of the International Conference on Emerging Trends in Computer and Electronics Engineering ICETCEE*, pages 13–18, March 2012.

A Khosravi, S Nahavandi, D Creighton, and AF Atiya. Lower upper bound estimation method forconstruction of neural network-based prediction intervals. *IEEE Tranactions on Neural Networks*, 22(3), March 2011.

C Nugteren and V Codreanu. CLTune: A Generic Auto-Tuner for OpenCL Kernels. In *MCSoC: 9th International Symposium on Embedded Multicore/Many-core Systems-on-Chip. IEEE*, 2012.

Harris Papadopoulos and Haris Haralambous. Reliable prediction intervals with regression neural networks. *Neural Networks*, 24(8):842–851, June 2011.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive Confidence Machines for Regression. In Elomaa T., Mannila H., and Toivonen H., editors, *Machine Learning: ECML 2002. ECML 2002. Lecture Notes in Computer Science*, volume 2430. Springer, Berlin, Heidelberg, 2002.

T Pearce, M Zaki, A Brintrup, and A Neely. High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. *arxiv*, 2018. URL https://arxiv.org/pdf/1802.07167.pdf.

Y Romano, E Patterson, and E Cands. Conformalized Quantile Regression. *arxiv*, 2019. URL https://arxiv.org/pdf/1905.03222.pdf.

V Vovk, V Fedorova, I Nouretdinov, and A Gammerman. Criteria of efficiency for conformal prediction. In *COPA 2016: Proceedings of the 5th International Symposium on Conformal and Probabilistic Prediction with Applications*, volume 9653, pages 23–29, April 2016.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer US, 2005.